UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/750,188	12/29/2000	Thomas S. Brady	3175-51	5835

21013 7590 04/14/2006

AGFA CORPORATION
LAW & PATENT DEPARTMENT
200 BALLARDVALE STREET
WILMINGTON, MA 01887

EXAMINER	
TUCKER, WESLEY J	
ART UNIT	**PAPER NUMBER**
2624	

DATE MAILED: 04/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/750,188	Applicant(s) BRADY, THOMAS S.
	Examiner Wes Tucker	Art Unit 2624

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 23 January 2006.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1,3-5,7,9-12,14,17,18,21 and 22 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1,3-5,7,9-12,14,17,18,21 and 22 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 01 June 2005 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Status of Claims

1. Applicants amendment filed January 23rd 2006 in response to the last office action has been entered and made of record.

2. Claims 1, 7 and14 are currently amended. Claims 2, 6, 8, 13, 15, 16, 19, 20 and 23-25 have been canceled. Claims 1, 3-5, 7, 9-12, 14, 17, 18, 21 and 22 are now pending.

Response To Arguments

3. Applicants arguments in view of the presented amendment have been considered but are not fully persuasive for at least the following reasons:

In Applicant's last response, applicant argued that Fallon did not teach that Run Length Encoding was not disclosed specifically for use with black and white data. Examiner cited the reference to Hamzy to show that RLE is exceedingly well known in the art for use with black and white data. Hamzy states explicitly that "Run-Length Encoding works best with black and white or cartoon style graphics" (column 6, lines 16-29). A 103 rejection was made from the combination of the Fallon and Hamzy.

Applicant now amends the independent claims to add the feature of ***wherein one predefined compression code represents white image data, and another predefined compression code represents black image data***.

From the previous discussion of Hamzy and Fallon, it is exceedingly clear that black and white data each have a predefined compression code. However, for clarification it will be further explained.

Run Length Encoding is performed by taking a character (or byte composed of eight bits) that describes a pixel, if that character occurs enough times in succession, then that sting of pixels or bytes are run-length encoding by taking the byte that describes the occurring pixel and coupling it with a number which describes the number of times that the reoccurring character or pixel occurs. This is how image data and indeed any data is run length encoded.

Fallon explicitly states that a dictionary of such characters is predefined giving 256 possible character codes (column 8, lines 5-10). A character that defines white data and a character that defines black data are no doubt included in a library defining possible pixel values in an image. Typically the white value of the grey scale is all 1s while the black value is all 0s. In black and white images, when there are only two color values to consider RLE is a very good compression option for the obvious reason that there will be many repeating characters in the image that can be optimally encoded vastly reducing the amount of data needed to describe the image.

As Hamzy teaches, run-length encoding works best with black and white graphics (column 6, line 24). Therefore, putting the teachings of Fallon and Hamzy together it can be seen that having a predefined compression code to represent black image data and a predefined compression code to represent white image data is inherent in RLE compression. Fallon teaches a library of predefined character or

compression codes for an image that would inherently contain a character code for white pixels and a character code for black data. Hamzy goes a step further to say that RLE works best with black and white data. The act of performing RLE compression on black and white data inherently requires that the black data have a code and that the white data have a code to be compressed. Therefore in the combination of Fallon and Hamzy it is exceedingly clear that RLE is performed on black and white data and that codes are required to represent the black and white data.

The previously presented rejection under 35 USC 103 in view of the combination of references to Fallon and Hamzy is maintained and is accordingly made FINAL.

Preface

4. The following is a brief discussion relating to how the Examiner views the Applicant's disclosed invention, vis-à-vis the so-called AGFA technique (AT). This is presented primarily to motivate the Prior Art rejections below.

Essentially, the AGFA technique is a straightforward combination of the well-known run-length encoding (RLE) and Lempel-Ziv (LZ) compression techniques. The details of the former were presented in the previous Office Action and the latter is discussed in the Applicant's *Background of the Invention*. LZ algorithms, and their variants, belong to a class of compression algorithms called dictionary compression algorithms. The effectiveness of LZ algorithms is known to degrade when confronted with long strings of redundant symbols. RLE, on the other hand, optimally compresses

such strings, though it has little utility in the compression of highly variable strings. Clearly, in this manner, RLE and LZ are complementary and, as such, seem well suited for combining. Indeed, combining the two compression methodologies is well known, as shown in Prior Art cited below.

The Applicant has chosen to extend this combined methodology (or limit its functionality, depending on one's point of view) by initializing the code dictionary to contain a set of predefined codes that are representative of white image data and black image data (and other "control" codes). Furthermore, the Applicant restricts the application of RLE to only white or black image data. These proposed modifications are straightforward and would have been obvious to one of ordinary skill in the art, particularly if one assumes some prior knowledge of the input image data.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 3-4, 7, 9-11, 14, 17-18, and 23-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over [Fallon03] (U.S. Patent 6,597,812) in view of Hamzy (U.S. Patent 6,711,294 to Hamzy et al.

The following is in regard to Claim 7. [Fallon03] disclose a method of lossless data compression, which exploits various characteristics of run-length encoding (RLE), parametric dictionary encoding, and bit packing ([Fallon03] Abstract). The method uses both predefined compression codes (e.g. the 259 entries of the initialized code dictionary (D)[1] – [Fallon03] column 8, lines 5-10 and column 6, lines 35-38) and compression codes defined during processing (e.g. "dynamically added" codes – [Fallon03] column 6, lines 40-43). The method comprises (refer, generally, to the example given in [Fallon03] column 10, lines 4-67 to column 11, lines 1-44):

(7.a_f.) Reading a first character[2] in an input sequence of characters (e.g. [Fallon03] Fig. 2A, step 203[3]).

(7.b_f.) Since all possible byte values are contained in the dictionary, the first character of the input sequence is assumed to correspond to one of the initial entries. Subsequent first characters[4] are examined ([Fallon03] column 10, lines 18-20). If a character does not match any of the entries in the code dictionary D, then that character is added to D and assigned a "compression code defined during

[1] These will be referred to as *initial entries*. According to [Fallon03] ([Fallon03] column 6, lines 35-38), 256 of the 259 initial entries (i.e. entries D[3]-D[258]) contain a character or byte corresponding to one of the possible values a single byte can represent. These are analogous to the predefined compression codes (PCC) of the Applicant's disclosure, in the sense that they are predefined and representative of common symbols or symbols expected to be present in the input data. Further notice that the dictionary of [Fallon03] contains reserved *control* codes ([Fallon03], column 6, lines 14-33) – e.g. a reset code (D[0]), a "run-length" code (D[1]), and an end code (D[3]). The similarity of these codes to those depicted in Figure 3 of the Applicant should be apparent.

[2] Any character in the input sequence (perhaps with the exception of the last) can be considered a first character, in the sense that any given character in the sequence is the first character of the sub-sequence of characters that immediately follow it. This is the manner in which "a first character" will be treated in this document.

[3] Notice that the caption in [Fallon03] Fig. 2A, step 203 reads "Read Next Input Byte". After initialization this "next" byte is actually the first byte of the string. See [Fallon03] column 8, lines 19-21. See also [4].

processing" (e.g. a dictionary index D[*i*]). Refer to [Fallon03] column 8, lines 50-55 and column 9, lines 1-10, 17-20, and 25-27.

(7.c_f.) Reading characters occurring immediately subsequent to the first character (e.g. the *next consecutive input bytes* – [Fallon03] Fig. 2A, step 204). Refer to [Fallon03] column 8, lines 23-35.

(7.d_f.) Determining that the next consecutive input bytes match the first character[4] in the input sequence of characters. Refer to [Fallon03] column 8, lines 23-35. See also [Fallon03] column 10, lines 15-29.

Note that (7.c_f.) and (7.d_f.) occur upon a determination that the first[4] character corresponds to an entry in the code dictionary. See [Fallon03] column 10, lines 9-29. Notice there that, after the first character, "a", has been read and determined to be a member of the code dictionary, the next consecutive input bytes (e.g. "b") are analyzed (7.c_f.) and, depending on the result, RLE (7.d_f.) is commenced. This is also apparent from [Fallon03] Figs. 2A-2B. Observe the flow of execution from step 211 to step 214 to step 202 and, finally, to step 205. Steps 211-212 encompass the "determination that the first[4] character corresponds to an entry in the code dictionary", while steps 204-205 encompass (7.c_f.)-(7.d_f.) above. From this perspective, steps 211-212 clearly precede steps 204-205 (notice the position of in [Fallon03] Fig. 2A). In [Fallon03], RLE proceeds B typically:

(7.e_f.) The first[4] character and the set of matching next consecutive input bytes are represented by an "output" code comprising ([Fallon03]

column 8, lines 33-39):

1. An RLE control code (i.e. D[1]).
2. A predefined compression code corresponding to the first character (e.g. the code word stored in the dictionary corresponding to the first[4] character, C).
3. The number of next consecutive input bytes matching the first[4] character (hereinafter, the *run-count*).

[Fallon03] teaches in the background section that any form of digital data to be compressed can benefit from the disclosed system and method including image data (column 1, lines 25-30 and column 2, lines 1-4). Therefore it would have been obvious to use the compression method to compress image data.

In view of the discussion form the previous office action entitled "Response to Arguments and Remarks", it should be clear that, although not disclosed, the compression method of [Fallon03] is applicable to image data. This is particularly true when one takes into account the prior applications of the method's component compression techniques – i.e. dictionary (LZ) compression and RLE – to images and image data. Indeed, one would expect the method of [Fallon03] to be effective in the compression of image data because, as demonstrated in [Fallon03] and pointed out above, the incorporation of RLE into LZ overcomes the shortcomings of LZ. Image data is known to consist of regions of uniform color and, therefore, contains strings of redundant bytes. As shown above, the method of [Fallon03] accommodates redundancy

well. Given this, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to apply the method of [Fallon03] to image data.

[Fallon03] does not explicitly disclose supplying predefined compression codes representative of black image data and white image data. Furthermore, [Fallon03] does not disclose the application of RLE specifically to strings of repeating black image data or white image data, as indicated in Claim 7. However it would have been obvious to one of ordinary skill in the art to use the RLE portion of the compression with particular focus for use in black and white portions of the image, as it is well known in the art that this is where RLE compression is most useful and effective. For the sake of argument [Fallon03] has been combined with U.S. Patent 6,711,294 to Hamzy et al., which teaches that "Run-Length Encoding works best with black and white or cartoon style graphics" (column 6, lines 16-29). Therefore it would have been obvious to one of ordinary skill in the art at the time of invention to use the compression method and system disclosed by Fallon with image data and more particularly that the RLE portion of the image compression operate specifically on the black and white image portions since Hamzy teaches RLE compression is optimal on black and white portions of the image.

As stated above, the code dictionary of [Fallon03] is initialized so as to contain symbols which are expected to be encountered in the input data stream. Areas of uniform black and uniform white may abound an image, depending on the type of image

(e.g. text or grayscale images). Furthermore, being the extremes of the color spectrum, these colors are common to nearly all images. Therefore, in an application of [Fallon03] to image data, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to provide entries in the code dictionary (predefined compression codes) corresponding to white image data symbols and black image data symbols, as such symbols are likely to be encountered in typical images. Because [Fallon03] performs RLE to sets of repeating symbols in the input data stream, areas of uniform black and uniform white would, in turn, be run-length encoded, where again the code consists of an RLE control code, a PCC (i.e. the dictionary index D[*i*] corresponding to either a white image data symbol or black image data symbol), and a run-count. Inherent to such an image-based application of [Fallon03] is the determination of whether a first[4] character represents either one of a white image data symbol or black image data symbol. This follows directly from step (7.b_f.) above.

Certain types of images are predominantly black and white (e.g. grayscale image, binary images, text, etc.). Clearly, the image data associated with such images consists primarily of black image data symbols and white image data symbols. In keeping with teachings of [Fallon03], one would naturally provide entries in the code dictionary that are representative of black image data symbols and white image data symbols, since these symbols are expected in the input image data. This was discussed previously. Moreover, to limit the size of the code dictionary one could advantageously restrict the code dictionary (aside from the aforementioned control codes) to include only entries corresponding to a black image data symbol and a white image data

symbol. Clearly, for predominantly black and white images, such a restriction has a negligible effect on the performance of the compressor. Therefore, in order to accommodate these types of images while providing a minimally sized code dictionary, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to restrict the code dictionary of an image-based implementation of [Fallon03] (aside from the aforementioned control codes) to initially include only entries corresponding to a black image data symbol and a white image data symbol.

In summary, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to apply [Fallon03] to image data – or more specifically, image data that is predominantly black and white – and, in order to accommodate such images efficiently, to restrict the code dictionary (aside from the aforementioned control codes) to entries corresponding to a black image data symbol and a white image data symbol. The resultant method would include:

(7.a.) Reading a first[4] character in an input sequence of characters representing an image. See step (7.a_f.) above.

(7.b.)
1. Determining whether the read first[4] character represents either one of a white image portion or a black image portion (i.e. determining whether the first character has a corresponding entry in the code dictionary – cf. [Fallon03] column 8, lines 53-55 and column 10, lines 15-22).
2. Upon the determination that the first[4] character does not

represent either of a white or black portion of the image (i.e. the first character has no corresponding entry in the code dictionary), representing the first character with an output sequence comprising a compression code defined during processing (cf. [Fallon03] column 9, lines 1-10, 17-20, and 25-27).

Upon determination that the first character does represent one of a white portion and a black portion of the image (i.e. the first character has a corresponding entry in the code dictionary – see steps (7.c_f.)- (7.e_f.)):

(7.c.) Reading characters occurring immediately subsequent to the first character in the sequence of characters.

(7.d.) Determining the number of repeated subsequent characters that match the read first character in the input sequence of characters.

(7.e.) Representing the first character and the determined number of repeated subsequent characters with an output sequence of characters comprising a PCC corresponding to the one of the white and the black portion of the image (i.e. the dictionary index D[*i*] corresponding to either a white image data symbol or black image data symbol – see above).

Applicant has amended the independent claims to add the feature of ***wherein one predefined compression code represents white image data, and another predefined compression code represents black image data***.

From the previous discussion of Hamzy and Fallon, it is exceedingly clear that black and white data each have a predefined compression code. However, for clarification it will be further explained.

Run Length Encoding is performed by taking a character (or byte composed of eight bits) that describes a pixel, if that character occurs enough times in succession, then that sting of pixels or bytes are run-length encoding by taking the byte that describes the occurring pixel and coupling it with a number which describes the number of times that the reoccurring character or pixel occurs. This is how image data and indeed any data is run length encoded.

Fallon explicitly states that a dictionary of such characters is predefined giving 256 possible character codes (column 8, lines 5-10). A character that defines white data and a character that defines black data are no doubt included in a library defining possible pixel values in an image. Typically the white value of the grey scale is all 1s while the black value is all 0s. In black and white images, when there are only two color values to consider RLE is a very good compression option for the obvious reason that there will be many repeating characters in the image that can be optimally encoded vastly reducing the amount of data needed to describe the image.

As Hamzy teaches, run-length encoding works best with black and white graphics (column 6, line 24). Therefore, putting the teachings of Fallon and Hamzy

together it can be seen that having a predefined compression code to represent black image data and a predefined compression code to represent white image data is inherent in RLE compression. Fallon teaches a library of predefined character or compression codes for an image that would inherently contain a character code for white pixels and a character code for black data. Hamzy goes a step further to say that RLE works best with black and white data. The act of performing RLE compression on black and white data inherently requires that the black data have a code and that the white data have a code to be compressed. Therefore in the combination of Fallon and Hamzy it is exceedingly clear that RLE is performed on black and white data and that codes are required to represent the black and white data.

The following is in regard to Claims 1. A hardware implementation of the compression method discussed above would represent an encoder and would inherently comprise a memory for storing the code dictionary (and, hence, the PCCs) and a processor configured so as to execute the aforementioned methodology. The substantive limitations of such an implementation have been addressed above with respect to Claim 7. For the sake of brevity, that discussion will not be repeated.

The following is in regard to Claim 14. A hardware implementation of the compression method discussed above would represent an imaging system, as such an implementation would entail the processing of image data. Such a system would inherently comprise a processor configured so as to execute the aforementioned

methodology. Because that processor would process image data (digital images are rasters), it can be considered a raster image processor. The substantive limitations of such an implementation have been addressed above with respect to Claim 7. For the sake of brevity, that discussion will not be repeated.

The compressed output sequence generated by such a processor serves little purpose in its compressed form. Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to provide the imaging system with a means to decode the compressed image data into its original form. Such a means can be reasonably viewed as an image controller, in the sense that, with the "raw", decompressed image data, it can be used to control various external raster devices (e.g. monitors, printers, etc.).

The following is in regard to Claims 3, 9, and 17. As shown above, the result of RLE in [Fallon03] is a code consisting of an RLE control code, a PCC (i.e. code corresponding to an initial entry representative of the first[4] character), and the run-count. Furthermore, it was shown that, with the straightforward modifications described above, the PCC resulting from RLE process is representative of either a black image data symbol or a white image data symbol.

In the compression method of [Fallon03], RLE commences only when it has been determined that the number of consecutive matching immediately following the first[4] character (i.e. the run-length) is greater than a threshold *s*. See [Fallon03] column 8,

lines 23-39. This effectively addresses the subject matter set forth in Claim 9. Analogous arguments can be made for Claims 3 and 17.

The following is in regard to Claim 10. The values in [Fallon03] is presumably predefined and, therefore, defined prior to the reading of the first[4] character in the input sequence.

The following is in regard to Claims 4, 11, and 18. As mentioned above, RLE according to [Fallon03] produces a code consisting of a run-count. Again, this value is indicative of the "number of characters in the matching one or more characters". Similar arguments apply to Claims 4 and 18.

The following is in regard to Claims 23-25. In [Fallon03], each byte of the input data is sequentially read (i.e. "substituting the next subsequent character in the input sequence [of bytes] for the first[4] character") and processed, until there are no more input bytes to process (i.e. the condition in [Fallon03] Fig. 2A, step 202 is not satisfied). This is apparent from the process loop depicted in [Fallon03] Figs. 2A and 2B. This effectively addresses the subject matter set forth in Claims 23-25.

6. Claims 5, 12, 21, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over [Fallon03], in view of Hamzy (U.S. Patent 6,711,294) and further in view of [Yellin98] (U.S. Patent 5,727,090).

The following is in regard to Claims 5 and 12. As shown above, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to modify [Fallon03] as taught by Hamzy so as to efficiently compress predominantly black and white image data. The result, as shown above, would have been a method satisfying the limitations of Claim 7. As stated above, the compressed output code, generated according to [Fallon03], consists of dictionary indices, D[*i*]. These serve as the code symbols composing the output code. Since dictionary D is limited to a fixed maximum size, indices D[*i*] are of a fixed bit-length (i.e. the constituent code symbols are of fixed bit-length). See [Fallon03] column 6, lines 48-51 and column 7, lines 29-40. Note that, in [Fallon03], the run-count also has a fixed bit-length, as implied in [Fallon03] column 13, lines 7-19 and 38-43[4]. [Fallon03], however, does not disclose the inclusion of a *continuation code* within the code symbols, or within the compressed output code.

[Yellin98] disclose a method for storing run-length encoded raster image data, wherein each run is indicated by a variable-length sequence of bytes with the repeated symbol expressed in a fixed-length field and the run-count in a variable-length field ([Yellin98] Abstract). With the exception of the last byte, each byte of the RLE-encoded code consists of a continuation code (or, *concatenation flag*, using the nomenclature of [Yellin98] – [Yellin98] column 2, lines 11-18; see also Fig. 1). Each run is represented by a sequence of bytes; the first byte includes the color from the color pattern (i.e. the

[4] Note that [Fallon03] treats both ("X" and "N") as words, each word presumably having the same length.

repeating input symbol) and, in the remaining space, the most significant bits of the run-count; the lower bits of the run-count trail into additional bytes whose continuation codes have been asserted. Refer to [Yellin98] column 2, lines 11-23; column 4, lines 44-55; column 5, lines 16-35; and Fig. 1.

Clearly, the usage of concatenation flags by [Yellin98] allows the run-count to occupy any number of bytes. As a result, very long runs of input symbols can be properly encoded. The concatenation flags further the provide RLE codes with readily distinguishable boundaries[5]. See [Yellin98] column 4, lines 51-55. Therefore, in order to support the compression of long runs of input symbols, it would have been obvious to one of ordinary skill in the art, at the time of the Applicant's claimed invention, to further modify the method of [Fallon03] so as to generate variable-length RLE codes, such as those of [Yellin98], which include a continuation code in each of the constituent bytes. The result is a method of compression that conforms to that of Claim 12. The rejection of Claim 5 follows similarly.

The following is in regard to Claims 21-22. As just discussed, the RLE codes, constructed according to [Yellin98], include a run-count that may occupy a variable number of bytes (i.e. a "multi-character value corresponding to the number of characters in the matching one or more characters"), wherein each of the constituent bytes includes a concatenation flag (i.e. a "continuation bit"). The RLE codes of [Yellin98] also consist of a PCC representative of the repeating color (i.e. the *color number* from the

[5] The boundaries are demarcated by an "OFF" concatenation flag.

color palette – [Yellin98] column 2, lines 19-20), though it would be understood that, in combination with [Fallon03], that PCC would be an initial entry[3] of the code dictionary representative of either a white image data symbol or a black image data symbol. This sufficiently addresses the subject matter set forth in Claims 21-22.

Conclusion

7. Applicant's amendment necessitated the new grounds of rejection presented in the Office Action. Accordingly, THIS ACTION IS MADE FINAL. See MPEP 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Wes Tucker whose telephone number is 571-272-7427. The examiner can normally be reached on 9AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bhavesh Mehta can be reached on 571-272-2214. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Wes Tucker

4-7-06

Matthew C. Bella

MATTHEW C. BELLA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600